|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Лабораторная работа № 10**

|  |  |
| --- | --- |
| **Тема** Алгоритм Плавающего горизонта построения трёхмерных поверхностей  **Студент** Якуба Д. В.  **Группа** ИУ7-43  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Куров А. В. |  |

Москва

2020 г.

Оглавление

[Цель работы 2](#_Toc42354349)

[Техническое задание 2](#_Toc42354350)

[Теоретическая часть 3](#_Toc42354351)

[Практическая часть 3](#_Toc42354352)

[Программная реализация алгоритма на ЯП Python 3](#_Toc42354353)

[Пользовательский интерфейс 3](#_Toc42354354)

[Демонстрация работы алгоритма 3](#_Toc42354355)

# Цель работы

Изучение и программная реализация алгоритма Плавающего горизонта построения трёхмерных поверхностей.

# Техническое задание

Должна быть разработана программа, позволяющая осуществлять ввод пределов и шага изменения координат x, z, выбора уравнения поверхности из заранее сформированного списка, построение поверхности. Должен быть обеспечен поворот изображения (поверхности) вокруг каждой из трёх координатных осей. Система координат должна быть неподвижной. Выполнить масштабирования для обеспечения размещения исходного изображения целиком в пределах поля вывода.

Список уравнений поверхностей задаётся в отдельном модуле.

# Теоретическая часть

## Трёхмерная графика

### Модели трёхмерных объектов

Модели являются отображением формы и размеров объектов. Основное назначение модели – правильно отображать форму и размеры определённого объекта.

В основном используются следующие три формы моделей:

1. Каркасная (проволочная) модель. В данной модели задаётся информация о вершинах и рёбрах объекта. Это одна из простейших форм задания модели, но он имеет один существенный недостаток: модель не всегда правильно передаёт представление о форме объекта.

2. Поверхностные модели. Это тип модели, который часто используется в Компьютерной Графике. Поверхность может описываться аналитически, либо задаваться другим способом (допустим, отдельными участками поверхности, задаваемыми в качестве участков поверхности того или иного вида). При этом вложенные криволинейные поверхности можно представлять в упрощённом виде, выполняя, например, полигональную аппроксимацию: такая поверхность будет задаваться в виде поверхности многогранника.

Данная форма также имеет свой недостаток: отсутствует информация о том, с какой стороны поверхности находится материал, а с какой стороны пустота. Это важно знать в том случае, если мы занимаемся компьютерным моделированием трёхмерных объектов.

Данный способ распространён в Компьютерной Графике, так как в данной стези нас не интересует, с какой стороны находится материал, мы оперируем поверхностями.

3. Объёмные модели. Отличие данной формы задания модели от поверхностной формы задания состоит в том, что в объёмных моделях к информации о поверхностях добавляется информация о том, где расположен материал. Это можно сделать путём указания направления внутренней нормали.

### Удаление невидимых линий поверхности

Решение данной задачи позволяет правильно представить то, как расположены поверхности в пространстве.

Данную задачу принято считать центральной задачей графики. Задача эта достаточно сложная и нет единого алгоритма её решения в силу того, что могут предъявляться разные требования к результату решения.

Решать поставленную задачу можно в объектном пространстве (в мировой системе координат) или в пространстве изображений (в экранных координатах). Точность решения в объектном пространстве выше, чем в пространстве изображения, так как в первом пространстве используются вещественные значения.

Можно считать, что в основе всех алгоритмов лежит сортировка объектов по глубине. Если в сцене объектов, то трудоёмкость будет оцениваться как:

Если же работа происходит в экранной системе координат, то трудоёмкость может оцениваться следующим образом:

Где – количество пикселей экрана. В этом случае от нас потребуется решении задачи о том, каким цветом высветить очередной пиксел. Для каждого пиксела потребуется найти ближайший к наблюдателю объект (найти объект с максимальным значением координаты ).

Таким образом, поскольку , то ответ на вопрос с какой системой работать ясен. Однако используя тот факт, что расположение по глубине объектов сцены изменяется только в точках, расположенных на линии пересечения поверхностей, можно сделать вывод о том, что использование пространства изображений окажется не хуже использования объектного пространства.

## Алгоритм Плавающего горизонта

Данный алгоритм предназначен для построения поверхности, заданных неявным уравнением .

Предположим, пусть имеется кусок некоторой заданной поверхности и от нас требуется изобразить её на экране.

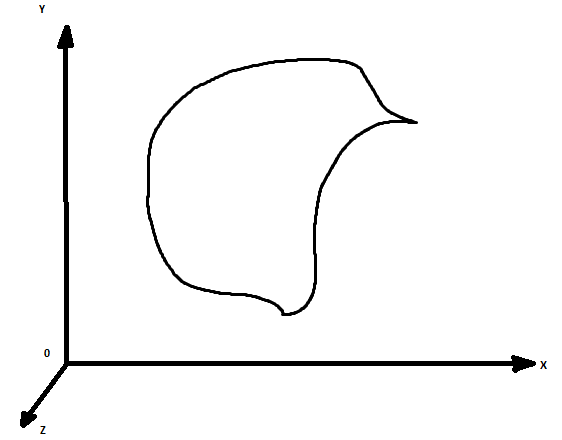


Рисунок , Пример

Наблюдатель располагается на положительной части оси и смотрит на начало координат.

Идея алгоритма состоит в том, что заданная поверхность рассекается плоскостями, перпендикулярными оси . То есть для начала от нас требуется с некоторым шагом расставить эти рассекающие плоскости.

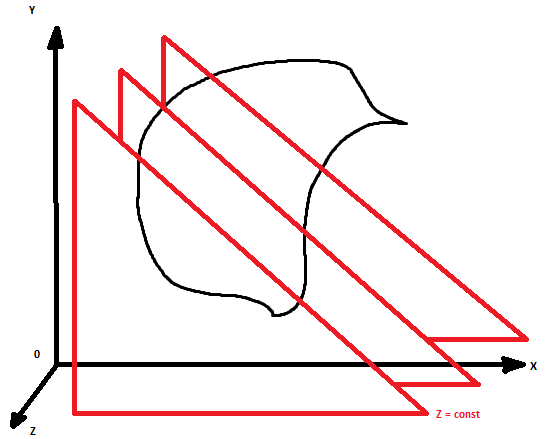


Рисунок , Рассечение

В отсечениях криволинейной поверхности и плоскостей будут получены некоторые кривые.

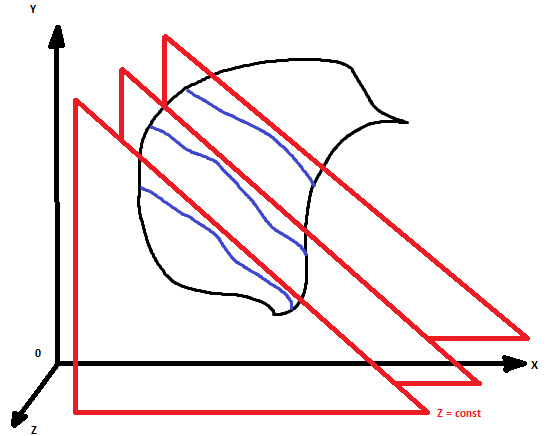


Рисунок , Результат отсечения

Зафиксируем основные этапы:

1. Рассматриваемая поверхность рассекается плоскостями, перпендикулярными оси .

В каждом отсечении получается кривая. Эта кривая, описываемая уравнением или .

2. Полученные кривые можно проецировать на плоскость (то есть на плоскость ) и изобразить видимые части каждой кривой.

Изображение строится начиная с кривой, полученной в ближайшем к наблюдателю сечении.

Кривая, полученная в сечении ближайшей плоскостью, является видимой.

Кривая, полученная во втором сечении, тоже будет видима. Связано это с тем, что вторая кривая расположена либо выше первой, либо ниже первой. В частом случае, когда они будут совпадать, будет получена одна кривая.

Начиная с третьей кривой понадобится решать задачу определения видимости точек кривой.

## Решение задачи определения видимости точек кривой

Поставленная задача решается в пространстве изображения.

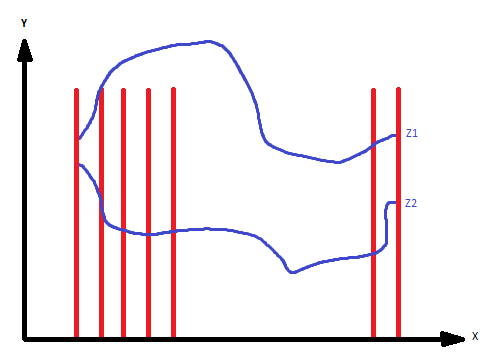


Рисунок , Пример

Берётся третья кривая и потребуется изображать те участки кривой, которые будут являться видимыми.

Участок кривой будет видим в том случае, если эта кривая располагается выше верхней кривой, либо ниже нижней кривой.

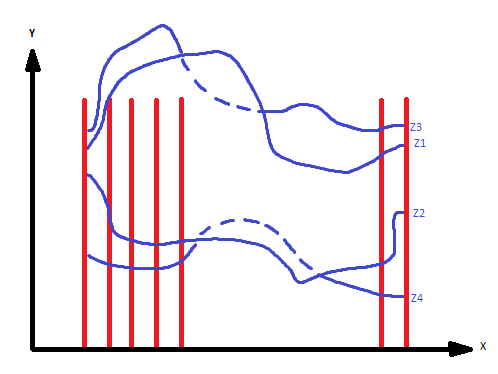


Рисунок , Добавление двух кривых с отображённой видимостью

Для того, чтобы решить поставленную задачу, потребуется вычислить функцию в очередной точке, расположенной на кривой, и сравнить значение вычисленной координаты с значением для раннее рассмотренных кривых. В том случае, если , то очередная точка, расположенная на текущей рассматриваемой кривой будет являться видимой.

# Практическая часть

## Программная реализация алгоритма на ЯП Python

## Пользовательский интерфейс

## Демонстрация работы алгоритма